M A D   8

Un Désassembleur pour 8008

Jérôme CHAILLOUX

ARTINFO/MUSINFO #28

MAD8 un desassembleur pour 8008

****
M A D 8
****

Jerome CHAILLOUX
Fevrier 1977

MAD8 est un desassembleur de rubans perfores hexadecimaux issus du micro-processeur 8008. Il permet d'obtenir des listages "en clair" de vos programmes a partir d'un ruban perfore. MAD8 fonctionne sur le T1600.

1.0 Les rubans hexadecimaux.

Les rubans hexadecimaux images-memoire sont produits au moyen des commandes W, E et N du moniteur 8008. Ces rubans ne contiennent que des caracteres imprimables et peuvent donc etre relus sur une TTY non connectee (TTY en mode local).

1.1 La commande N (null command)

syntaxe : .N

perfore une avance bande de 60 caracteres nulls (code 00).

1.2 la commande W (write command)

syntaxe : .W adresse de debut , adresse de fin

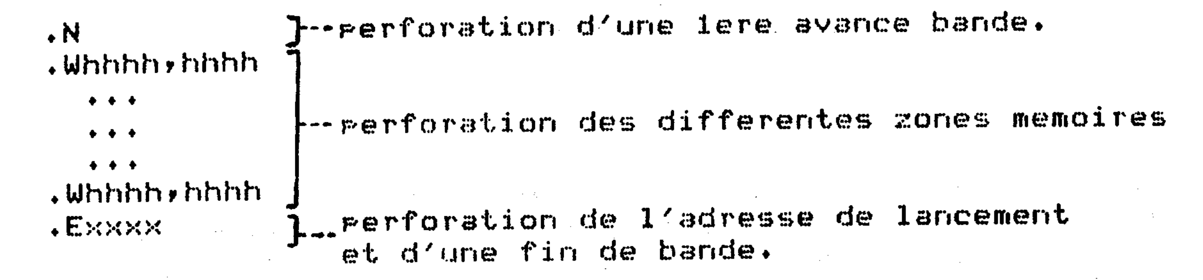
perfore (dans le format decrit ci-dessous) l'image le la zone memoire commencant et se terminant aux adresses specifiees dans la commande. On peut emettre plusieurs commandes W a la suite pour obtenir, sur le meme ruban physique, les images de zones memoires non-contigües.

1.3 la commande E (end-of-file commande)

syntaxe : .E adresse de lancement

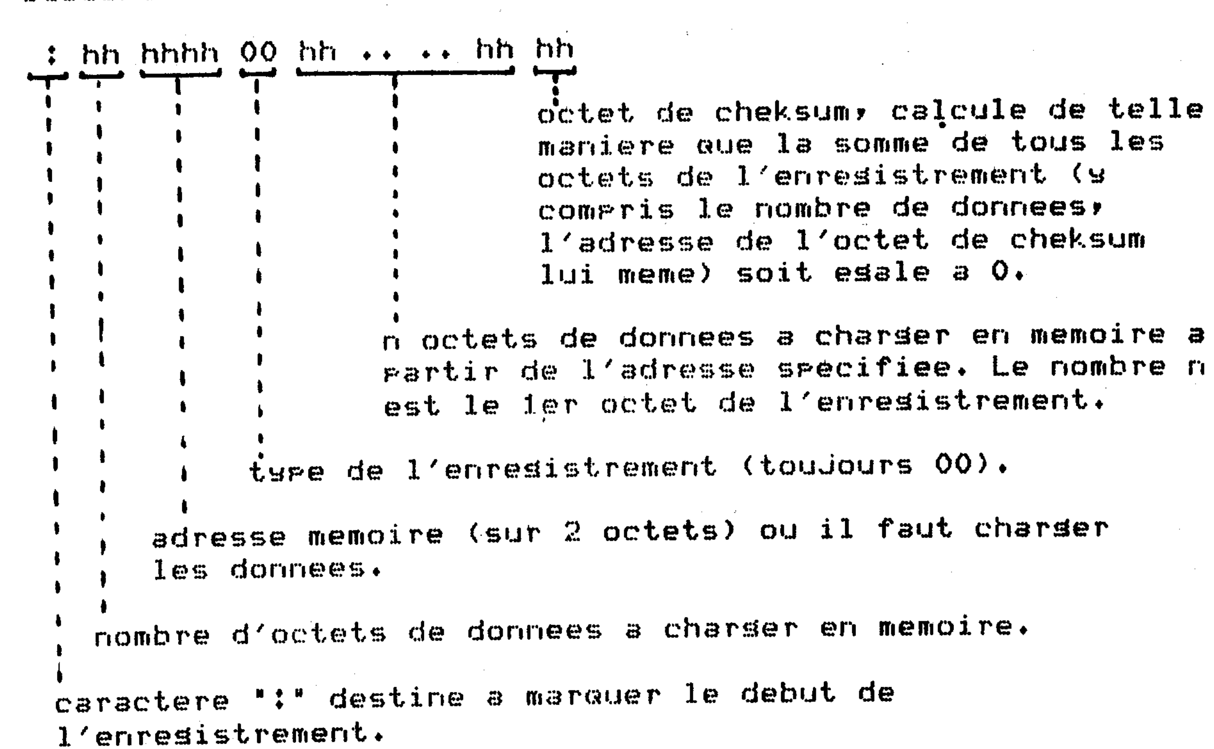
perfore un bloc "fin de ruban" (voir son format ci-dessous) qui contient l'adresse de lancement du programme. Cette commande perfore en outre une avance bande.

1.4 enchainement des commandes de perforation.



1.5 format des rubans hexadecimaux

Ils sont formes d'enregistrements qui possedent tous la meme structure. Chaque octet est represente par deux digits hexadecimaux (hh).



La fin logique du ruban est detectee par un enregistrement dont le nombre de donnees a charger est egal a zero.

```
 : 00 hhhh 00 hh
 |  |   |   |  |
 |  |   |   |  octet de cheksum
 |  |   |   toujours 00.
 |  |   adresse de lancement du programme.
 |  marque de fin de ruban logique.
 marque de debut d'enregistrement.
```

2.0 Principe du desassembleur.

Les rubans hexadecimaux sont tres utiles pour garder des programmes sous un format tres reduit, en revanche, ils sont peu commodes pour documenter, modifier ou meme comprendre ces memes programmes. MAD8 va editer chaque octet code du ruban perfore sous les differents formats :
- mnemonique instruction (INTELGREU),
- valeur hexadecimale
- caractere ASCII.

Les instructions du 8008 etant de longeur variable, les resulats ainsi obtenus sont tres mediocres pour le listage des instructions. Par exemple le desassemblage des 3 octets correspondant a l'instruction JMP 1000 produit l'edition de :

```
JMP  44  D
HLT  00  ?
INC  10  ?
```

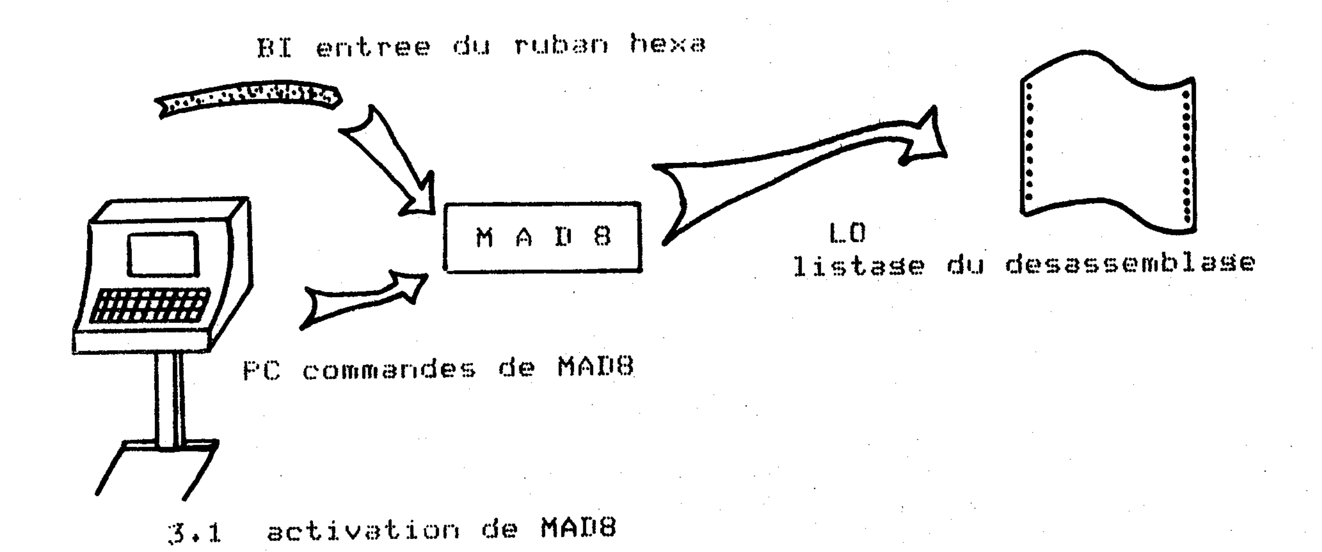
MAD8 va s'efforcer de distinguer dans l'image memoire les zones contenant des instructions, des zones contenant des donnees. Pour ce faire, MAD8 va simuler l'execution du programme a partir de son adresse de lancement, et marquer les octets qui sont consideres comme des instructions. Cette simulation consiste a interpreter les instructions qui provoque des ruptures de sequence i.e. les instructions de type HLT, JMP, CAL, RET et RST, conditionnelles ou inconditionnelles. Ce marquage ne peut s'effectuer qu'apres avoir charge en entier le programme a desassembler.

Une fois le marquage effectue, MAD8 edite l'image memoire sous les formats instructions ou donnees.

Ce principe a toutefois des limitations. Les branchements indirects et/ou indexes ne peuvent pas etre interpretes. Les donnees ne doivent pas etre melangees au programme. En particulier les appels de sous-programmes dont les arguments sont places juste en dessous de l'appel effectif risque de placer MAD8 dans un etat de confusion deplorable.

3.0 utilisation du desassembleur.

MAD8 est un utilitaire standard du T1600 qui utilise les differentes FUs :



3.1 activation de MAD8

*CALL MAD8 appel du programme MAD8. Si BOS/D imprime le message d'erreur ERR 06, le programme n'est plus sur le disque; il ne vous reste plus qu'a desassembler votre programme a a la main ou a remettre MAD8 sur le disque.

*BI TR affectation du lecteur de ruban

*PC TK affectation du clavier TTY pour entrer les commandes de MAD8.

*LO LP affectation de l'imprimante pour le listage final.

3.2 commandes MAD8

*IMAD initialise MAD8 et lit le 1er ruban sur l'unite BI. Cette commande est obligatoire et ne doit etre emise qu'une seule fois sous peine de perdre l'image memoire qui avait ete cree.

ou

MAD8 ne peut simuler que les 8 1ers k du 8008. Si vous voulez desassembler des programmes en REPROM (i.e des programmes dont l'adresse est plus grande que 2000 hexa), vous pouvez specifier dans la commande la 1ere adresse a simuler.

*IMAD,hhhh Cette deuxieme forme n'est dons pas a utiliser pour des programmes en RAM.

*CMAD permet de lire d'autres rubans sans reinitialiser le systeme, si votre votre programme se trouve sur plusieurs rubans perfores. Cette commande peut etre emise plusieurs fois.

*MMAD marque l'image memoire. On suppose que la premiere adresse du programme (son adresse de lancement) a ete lue sur le bloc fin de ruban du dernier ruban lu.

*MMAD,hhhh marque l'image memoire a partir de l'adresse specifiee dans la commande. Cette commande peut etre emise plusieurs fois en particulier pour specifier les differentes adresses se trouvant dans une table de branchements indirects indexes.

*LMAD edite sur l'unite LO le resultat du desassemblage. Cette commande peut etre emise plusieurs fois pour obtenir plusieurs copies du desassemblage.

*EOJ fin d'execution de MAD8.

3.3 utilisation du disque

Le lecteur de ruban de la TTY est tres lent. Il est parfois avantageux de creer un fichier sur disque contenant l'image du ruban hexadecimal a desassembler ce qui evite de recharger le ruban apres chaque erreur.
Pour copier le ruban perfore sur disque, il faut utiliser l'utilitaire standard du T1600 : le FUP6.

*CALL FUP6 appel de l'utilitaire FUP6.

*INPUT,TR,PTAP definition du support d'entree

```
*OUTPUT,nom-:I definition du fichier de sortie. L'extension
              :I est reservee pour les fichiers hexa de
              l'Intel.

*TRANSF       effectue le transfert

*EOJ          fin du travail
```

4.0 Exemples d'utilisation de MAD8

desassemblage direct d'un ruban perfore :

```
*CALL MAD8         appel de MAD8.
*BI TR             selection du lecteur en entree.
*LO LP             selection de l'imprimante.
*IMAD              initialisation et lecture
 --- lecture du ruban ---
*MMAD              marquage des instructions.
*LMAD              listage de l'image memoire.
 --- impression du resultat ---
*EOJ               voila le travail
```

Exemple du desassemblage complet du moniteur 8008.

```
*EOJ

*CALL FUP6         creation d'un fichier disque
*INPUT,TR,PTAP     contenant le ruban hexa du moniteur.
*OUTPUT,MONIT-:I,D2
*TRANSF
 --- lecture du ruban ---
*EOJ

*CALL MAD8
*BI MONIT-:I,D2
*PC TK
*LO LP
*IMAD,2000
*MMAD
*MMAD,38A3         marquage de tous les modules
*MMAD,39DE         du moniteur.
*MMAD,3967
*MMAD,39A9
*MMAD,39D4
*MMAD,3A00
*MMAD,3A13
*MMAD,3A1A
*MMAD,3C43
```

```
*MMAD,3A45
*MMAD,3A5F
*MMAD,3A8B
*MMAD,3A91
*MMAD,3AF6
*MMAD,3B68
*MMAD,3BB7
*MMAD,3BD5
*LMAD
 --- impression du desassemblage ---
*EOJ
```

5.0 Exemple de listage produit par MAD8.

```
---
---
0040  JMP 0703
0043  LLI 32 2
0045  LHI 00
0047  LCI 01
0049  JMP 3D80
004C  LLI 30 0
004E  LHI 00
0050  LAM
0051  INL
0052  LBM
0053  LHA
0054  LLB
0055  RET
0056  CAL 004C
0059  LME
005A  CAL 3DEB
005D  LAH
005E  LBL
005F  LLI 30 0
0061  LHI 00
0063  LMA
0064  INL
0065  LMB
0066  RET
0067             IN  41 A
0068             CFC 42 B
0069             IN  43 C
006A             JMP 44 D
006B             IN  45 E
006C             JFZ 48 H
006D             --- 4C L
006E             IN  4D M
006F             IN  49 I
0070  CAL 3F44
0073  LBI 08
0075  LLI 6F
0077  LHI 00

0079  CPM
007A  JFZ 007F
007D  LAB
007E  RET
007F  DCL
0080  DCB
0081  JFS 0079
0084  JMP 3C43
0087  LLI 33 3
0089  LHI 00
008B  LME
008C  RET
008D  CPI 46 F
008F  RTZ
0090  CPI 54 T
0092  JFZ 3C43
0095  LAI 20
0097  LLI 33 3
0099  LHI 00
009B  ADM
009C  LMA
009D  RET
009E  CAL 00A7
00A1  LLI 33 3
00A3  LHI 00
00A5  LEM
00A6  RET
00A7  CPI 43 C
00A9  RTZ
00AA  CPI 5A Z
00AC  LBI 08
00AE  JTZ 00BF
00B1  LBI 10
00B3  CPI 53 S
00B5  JTZ 00BF
00B8  LBI 18
00BA  CPI 50 P
00BC  JFZ 3C43
00BF  LAB
00C0  JMP 0097
00C3  CPI 08
00C5  JTZ 055E
```

```
00C8  ADE
00C9  ADI  7C
00CB  JMP  0553

----

0500  CAL  3CC7
0503  LBI  3A :
0505  CAL  3809
0508  CAL  0043
050B  LLI  30 0
050D  LMD
050E  INL
050F  LME
0510  CAL  3CC7
0513  CAL  004C
0516  CAL  3DFB
0519  CAL  3F44
051C  CPI  24 $
051E  JTZ  3844
0521  CPI  42 B
0523  JFZ  0538
0526  CAL  3F44
0529  CAL  3F44
052C  CAL  3C4B
052F  CAL  0043
0532  CAL  0056
0535  JMP  0510
 :     :          :
```

c'est probablement des données

c'est probablement du programme

c'est sûrement l'adresse d'implantation